

Construction d'un OS pour système embarqué

1.
 1. Objectif du module
 2. Equipe pédagogique et organisation du module
 3. Demarche
 4. TME

Objectif du module

L'objectif de ce module est d'analyser en détail le fonctionnement d'un système d'exploitation pour système embarqué. Le cours rappelle brièvement les principaux concepts des systèmes d'exploitation, puis détaille l'implémentation d'un système d'exploitation spécifique pour des architectures matérielles de type SOC (system on chip), c'est à dire constituées d'un ou plusieurs processeurs 32 bits, de mémoire embarquée, et de différents contrôleurs de périphériques mappés en mémoire. On vise donc des systèmes embarqués autonomes capables de contrôler un équipement industriel.

Equipe pédagogique et organisation du module

Les cours ont lieu en salle 42-43 310 le vendredi de 13h30 à 15h30. Les TME ont lieu les jeudi de 13h30 à 17h15 en salle 65-66 406.

- Franck Wajsburt
- Ghassan Almaless

Demarche

Le cours est organisé autour de la réalisation pratique d'un OS embarqué multi-tâches compatible POSIX.

Les TME consistent à écrire, à partir de rien ou presque, les différents composants du système d'exploitation. Celui-ci est mis au point en simulation sur un SOC modélisé en SystemC en utilisant la plate-forme de modélisation SoCLib, et composé de 4 processeurs mips32, d'une mémoire, d'un timer, d'un multi-tty (terminal), d'un contrôleur video, d'un contrôleur de disque et d'un dma. La progression que vous allez suivre :

- Un noyau d'OS qui virtualise le processeur, sans communication entre les tâches, sans partage de devices mais avec une HAL.
- Ajout des communication intertâches, partage de devices: mécanisme de file d'attente et événements.
- user land : ABI kernel (liste des syscall) + pthread + dietlibc + compilation séparée.
- Gestion des fichiers, buffer cache.

Le tableau ci-après définit le placement des mémoires et les périphériques dans l'espace d'adressage physique.

Périphériques	Segments dans la ROM	Segments dans la RAM
---------------	----------------------	----------------------

TME

- Reset
- Bootloader?
- kentry
- heap_manager
- hal_cpu hal_arch